NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

Optical connection in GMPLS Lightwave Switching Simulator (GLASS)

Version: Draft 1.0

TABLE OF CONTENTS

1 INTRODUCTION

This document presents the structure of the optical connections used in the GLASS framework. The connections are used as input for the algorithms (routing and wavelength assignment) to compute the routes and light paths between two optical nodes [1]. The algorithms can be static (no simulation time) or dynamic (message exchanges during the simulation).

2 GENERAL STRUCTURE

This section introduces the structure used in GLASS to store the information about the connections, the routes, and the lightpath. All the classes related to the connections are located in the package gov.nist.antd.optical.path.

First the process of creation of an optical connection is as follow:

- Creation of an optical connection object and its quality of service.
- Run of a routing algorithm that will try to compute one or more possible routes according to the quality of service requested.
- If the routing is done, then running of a wavelength algorithm to create a lightpath and reserve the resources (lambdas).

GLASS also provides the use of Routing and Wavelength Assignment (RWA) algorithms that creates the route and the lightpath in one step. For more information about the algorithm see [2].

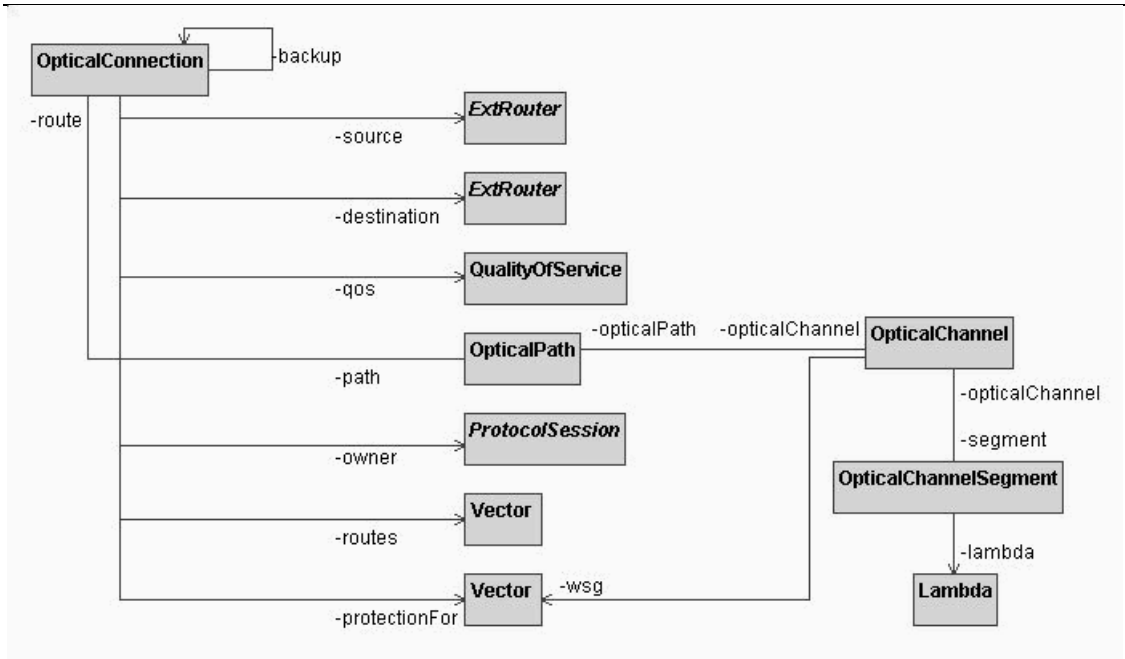


Figure 1 Overview of the connection's structure

The section 4 presents each component of the connection in detail.

3 STORAGE OF CONNECTIONS

In the glass framework, all the connections are stored in path containers to facilitate their management and to find them during the simulation.

There are multiple path containers per net, defined by their name (usually the name of the routing algorithm of the connection).

The path container is defined as follow:

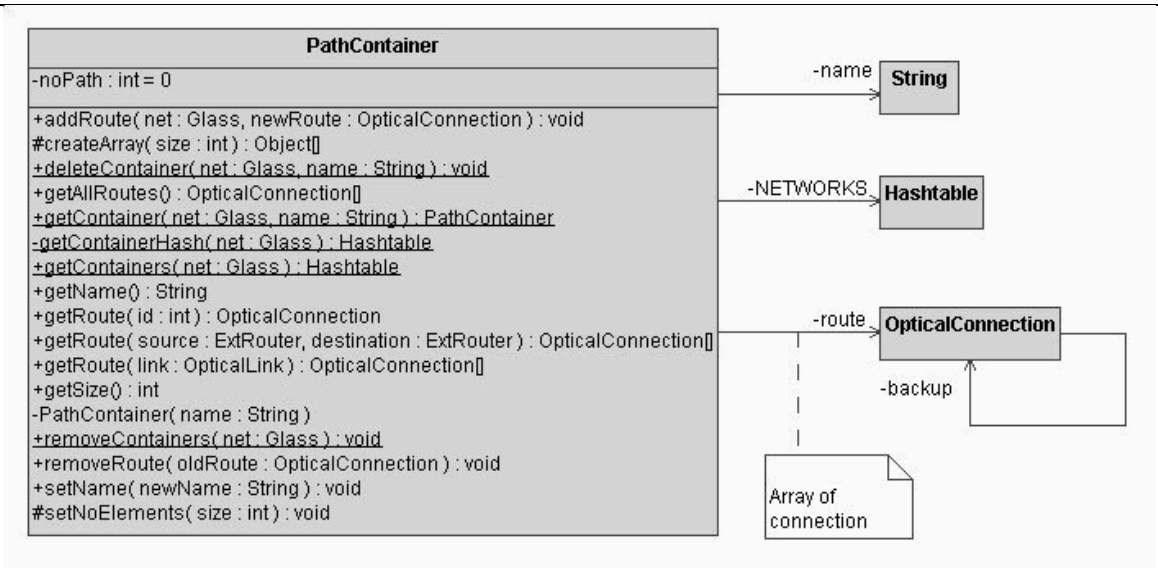


Figure 2 UML diagram of PathContainer

The Hashtable NETWORKS is used to separate the path containers of different networks. This allows having multiple path containers with the same name but in different network.

The name identifies the path container in the net.

The path container stores a list of route that can be retrieved according to the connection id (unique in the net), the source and destination node, a specific link or also all the connections contained in the path container.

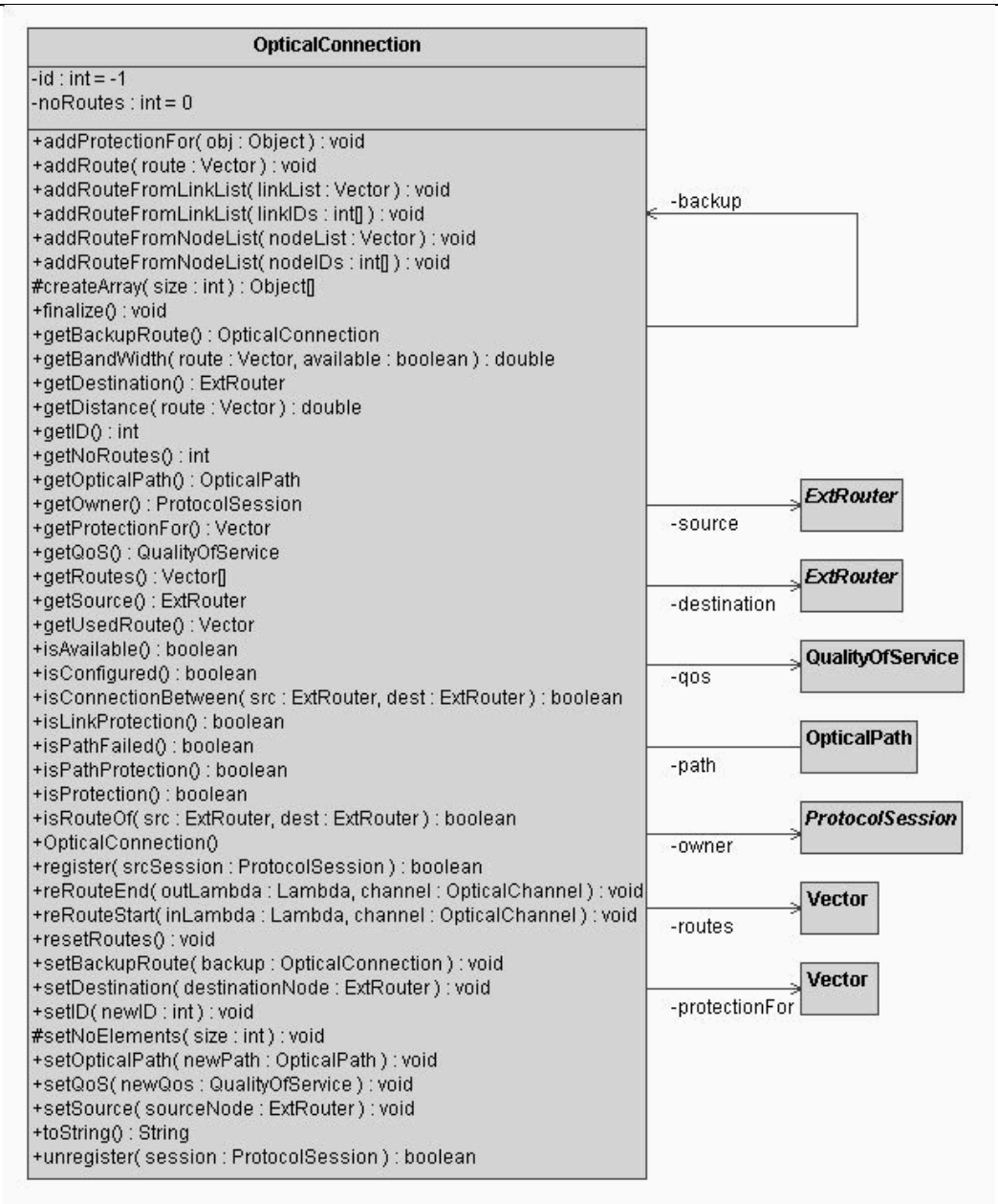
4 IMPLEMENTATION DETAILS

This section goes in details on each component that compose the connection:

- OpticalConnection
- OpticalPath
- OpticalChannel
- OpticalChannelSegment

4.1 OPTICAL CONNECTION

The OpticalConnection is the first element created in order to setup a connection between two nodes. Its configuration takes an important place in the simulation.



The *source* is the optical node that is trying to connection to the *destination*. This connection has some constraints defined in the *QoS* (see 4.1.1 for explanation of the QualityOfService).

The result of a routing algorithm on a connection is that creation of one or more possible routes, stores in the Vector *routes* (see for explanation of the structure of the routes). Once the routing algorithm has run, the wavelength assignment algorithm is using the information of the routes to create a *path*. This lightpath is defined in GLASS by the class OpticalPath.

One goal of GLASS is to also help on the study of fault/restoration algorithm; the connection also provides backup information. A connection can be protected (then it has a *backup*) or protect a set of connections (attribute *protectionFor*).

Finally, before a connection can be used, a protocol must register to the path. This also allows the framework to make sure that there is only one user at a specific time for a route. Once a protocol registers to a connection, it becomes the *owner*.

4.1.1 THE QUALITY OF SERVICE

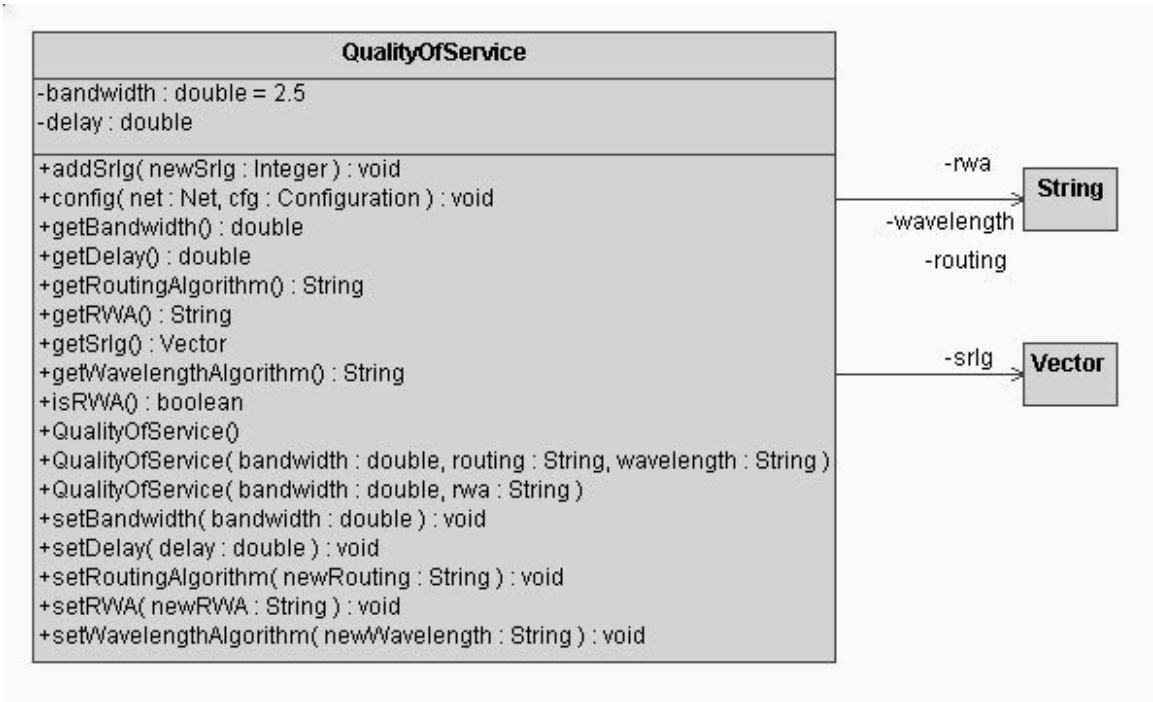


Figure 3 UML diagram of QualityOfService

The **QualityOfService** represents the constraints for the algorithms on the path in order to satisfy the requested connection. It also contains the name of the algorithms that must be used (*routing*, *wavelength* and *RWA*). The constraints are the minimum bandwidth of the connection, the maximum delay and a more complex one, the Shared Risk Link Group (SRLG).

The use of the attribute depends on the algorithm. For example, the ShortestPathDistance does not consider the delay of the SRLG to compute the route, but the ShortestPathSRLG consider the SRLG (but not the delay).

It is also not possible to request a routing algorithm and a wavelength algorithm in the same time as an RWA algorithm, because there would be a conflict.

4.1.2 THE ROUTE STRUCTURE

The route is specified by the links that the lightpath can go. As it is possible to have multiple links between two nodes, GLASS framework uses a mechanism of bundle. The class gov.nist.optical.util.PtPBundle represents the bundle of links.

The following example shows how the bundle works.

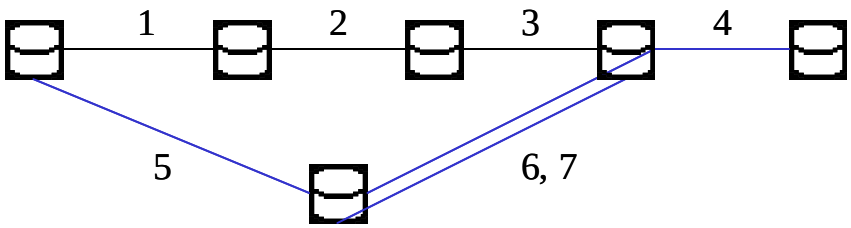


Figure 4 Example a route

In this example, let's suppose that a routing algorithm as found that the optimum path is the blue path. Then at the second stage, there are two possible links. The internal structure that needs to be created is a Vector of PtPBundle; in this case 3 elements must be created.

- In the first one, put the link 5,
- In the second one, put the link 6 and 7,
- And in the last one, the link 4.

The wavelength algorithms are working on this internal structure to find the path.

Note: GLASS provides tools to help the creation of the route structure in the class PathUtil (see [3] for more information).

4.2 THE OPTICAL PATH STRUCTURE

The optical path is the second step in to succeed the creation of a connection. This section shows the structure that is created by the wavelength algorithm.

4.2.1 THE OPTICALPATH

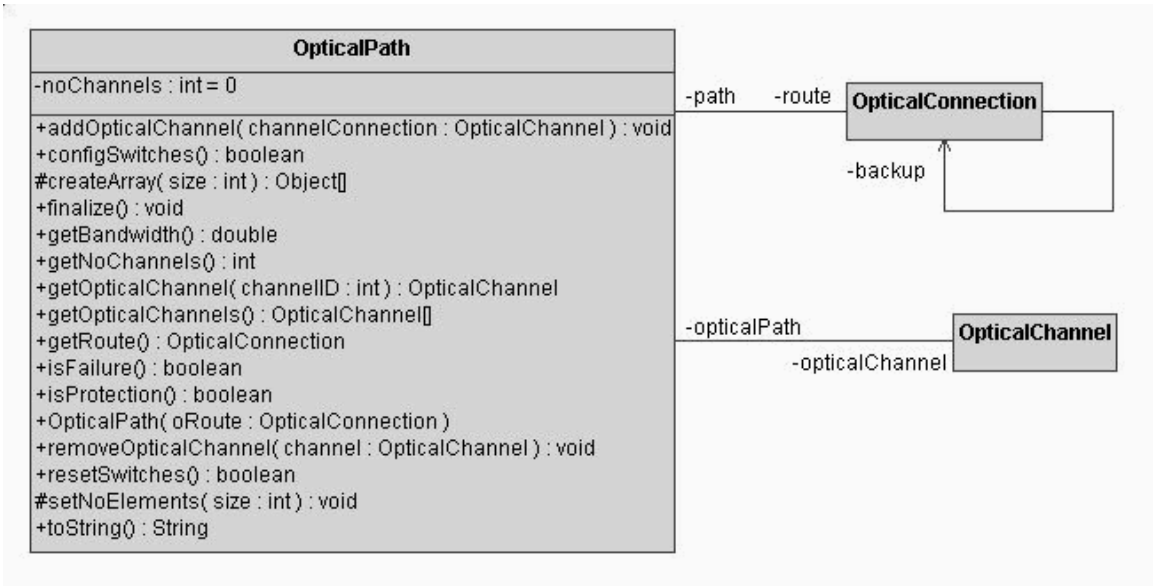


Figure 5 UML of OpticalPath

There is a double linkage between the OpticalConnection and the OpticalPath so that it makes easier to go through the components of the connection.

The OpticalPath is composed of a set of OpticalChannel. Each of the channels is an end-to-end path from the source to the destination. It is possible to have more than one channel in order to have a higher bandwidth. The class also provides the methods to add/remove channels.

4.2.2 THE OPTICALCHANNEL AND OPTICALCHANNELSEGMENT

As mentioned before, the OpticalChannel is an end-to-end connection. It is composed of segment. Each of the segments represents the lambda that has been reserved by the wavelength algorithm for each link that composes the route.

Figure 6 UML diagram of OpticalChannel and OpticalChannelSegment

The role of a wavelength algorithm (or the wavelength part of a RWA algorithm) is to find the wavelength that can be used in all the links of the route computed by the routing algorithm, and to create the previous structure.

5 USING THE COMPUTED CONNECTIONS

Once the route and the lightpath have been computed, the resources have been reserved. But it is not enough to use a connection. The switches must also be setup along the path.

To do so, 2 methods can be used:

- Instantaneous setup of the switches. In this case, a call one of the method connectSwitches () in the class ConnectionUtil (see [3]) will connection the lambdas together at the intermediate nodes and also connect the add ports at the source and the drop ports at the destination so that the connection is ready to use (except if there is an error while configuring the switches).
- Dynamic setup of the switches. GLASS allows a dynamic configuration by sending messages to the nodes that compose the path in order to setup the switches (even the route and path might also have been dynamically computed).

A protocol has to register to the connection in order to get the right to send data.

References

[1] Nodes in GMPLS Lightwave Switching Simulator (GLASS)
By NIST/ANTD
URL: http://www.antd.nist.gov/glass

[2] Algorithms in GMPLS Lightwave Switching Simulator (GLASS)
By NIST/ANTD
URL: http://www.antd.nist.gov/glass

[3] Utils in GMPLS Lightwave Switching Simulator (GLASS)
By NIST/ANTD
URL: http://www.antd.nist.gov/glass